

**REMARKS**

Claims 1-29 were pending in the application. Claims 1-7, 12-18 and 23-29 were rejected under 35 U.S.C. §102(a) as being anticipated by Kamel et al. Claims 8-11 and 19-22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kamel as applied to claims 1-7, in view of Richter et al. Claims 1, 3-5, 7, 12, 14-16, 18, 23, 25-27 and 29 have been amended. Claims 2, 13 and 24 have been canceled, leaving claims 1, 3-12, 14-23, and 25-29 presently pending in the application. Reconsideration and reexamination of the application in view of the amendments and following remarks is respectfully requested.

The present invention is directed to managing a shared read/write buffer pool and the execution of read and write commands to ensure that free blocks are available to temporarily store read or write data. A read command typically results in a number of pending read data requests that are satisfied over time, each of which requires the use of blocks in the shared read/write buffer pool. Without the management provided by the present invention, during the pendency of the read command, write commands may be initiated that consume the remaining buffer pool resources such that there will not be enough buffer pool resources to satisfy the pending read data requests. With both the pending read data requests and pending write commands consuming all of the buffer pool resources, neither may be able to complete, and a lockup condition may occur. To reduce the amount of read data re-transmissions, the write command may be throttled based upon the amount of pending read data requests that are currently unsatisfied and the amount of free blocks available. If the currently available free blocks would be substantially consumed by the total outstanding inbound read data requested, no more write commands will be initiated. As inbound read data is received into allocated buffers and transferred to the initiator device, the blocks in the buffer pool are freed up. When the read data transfer is completed or sufficient buffer resources have been freed up, the write data command may be initiated.

**Claims 1-7, 12-18 and 23-29 were rejected under 35 U.S.C. §102(a) as being anticipated by Kamel.** Of these, claims 1, 3-5, 7, 12, 14-16, 18, 23, 25-27 and 29 have been amended. In particular, claim 1 has been amended to include the following limitations:

- (1) a shared read/write buffer pool of blocks for temporarily storing write data to be sent to a peer device and read data received from the peer device;
- (2) a receive list memory which contains descriptor pointers to free blocks and blocks filled with read data in the shared read/write buffer pool;
- (3) a transmit list memory which contains descriptor pointers to free blocks and blocks filled with write data in the shared read/write buffer pool; and
- a processor programmed for
- (4) determining a number of blocks in the shared read/write buffer pool that will be required to store the read or write data for the pending read data requests and the new write data command,
- (5) determining a number of free blocks in the shared read/write buffer pool, and
- (6) throttling the new write data command if the number of free blocks in the shared read/write buffer pool is insufficient to store the read data for the pending read data requests and the write data for the new write data command.

Claims 12 and 23 have been similarly amended to include the following limitations:

- (1) temporarily storing write data to be sent to a peer device and read data received from the peer device in a shared read/write buffer pool of blocks;
- (2) storing descriptor pointers to free blocks and blocks filled with read data in the shared read/write buffer pool in a receive list memory;
- (3) storing descriptor pointers to free blocks and blocks filled with write data in the shared read/write buffer pool in a transmit list memory;
- (4) determining a number of blocks in the shared read/write buffer pool that will be required to store the read or write data for the pending read data requests and the new write data command,
- (5) determining a number of free blocks in the shared read/write buffer pool, and
- (6) throttling the new write data command if the number of free blocks in the shared read/write buffer pool is insufficient to store the read data for the pending read data requests and the write data for the new write data command.

Because limitations (1) though (6) are similar in claims 1, 12 and 23, they will be addressed together. Claims 2, 13, and 24 have been canceled, rendering the rejection of those claims moot. With regard to claims 1, 3-7, 12, 14-18, 23 and 25-29, in view of the current amendments, it is respectfully submitted that the rejection of those claims has been overcome.

Kamel fails to disclose, teach or suggest these limitations. In particular, Kamel fails to disclose, teach or suggest *storing both read data and write data in a shared read/write buffer pool*, as recited in limitation (1) of claims 1, 12 and 23. Kamel only discloses a *write buffer pool 28* for storing data to be written to the disk (col. 5 lines 22-24). Note that read data is retrieved directly

from the disk (col. 4 lines 42-48), and is not stored in the write buffer pool 28. Although the Examiner states in the Office Action that col. 2 lines 50-52 disclose a shared read/write buffer pool, those lines only disclose that read and write *requests* are stored on the buffer pool. Read and write *data* are different from read and write *requests*. A request is not data.

Kamel also fails to disclose *storing descriptor pointers to free blocks and blocks filled with read data in the shared read/write buffer pool in a receive list memory, and storing descriptor pointers to free blocks and blocks filled with write data in the shared read/write buffer pool in a transmit list memory*, as recited in limitations (2) and (3) of claims 1, 12 and 23. First and foremost, as demonstrated above, Kamel does not disclose a shared read/write buffer pool. Second, Kamel only discloses a Sequence Control Broker (SCB) that stores an ordered list of pointers to all media segment file blocks of a video stream to be retrieved (i.e. *pointers to read data*) (col. 3 lines 53-56). However, Kamel contains no disclosure at all related to pointers to write data and free blocks in a shared read/write buffer pool in a transmit list memory, or pointers to free blocks in the shared read/write buffer pool in a receive list memory. Although the Examiner states in the Office Action that col. 6 lines 65-66 disclose a receive list memory which contains descriptor pointers to free blocks, those lines only mention that write requests are stored in the write buffer pool. Furthermore, although the number of free buffer slots in the write buffer pool,  $n_f(t)$ , is described and used in a computation, Kamel does not disclose pointers to those free buffer slots.

Kamel further fails to disclose *determining a number of blocks in the shared read/write buffer pool that will be required to store the read or write data for the pending read data requests and the new write data command*, as recited in limitation (4) of claims 1, 12 and 23. Again, the fundamental argument is that Kamel does not disclose a shared read/write buffer pool. The Examiner cites col. 7 lines 1-8 as disclosing this limitation, but those lines only describe the computation of an estimated deadline for processing a *write request*, which will occur when the write buffer pool fills up with subsequent *write data*. Note that any mention of determining the number of blocks needed to *store the read data from pending read data requests* in the write buffer pool is entirely absent from the discussion, which is not surprising, because the write buffer pool does not store read data.

Kamel also contains no disclosure at all related to *determining a number of free blocks in the shared read/write buffer pool*, as recited in limitation (5) of claims 1, 12 and 23. Once again, Kamel does not disclose a shared read/write buffer pool. The Examiner cites col. 6 lines 66-67 as disclosing this limitation, but although Kamel assigns a parameter to the number of free buffer slots in the write buffer pool,  $n_f(t)$ , and uses it in a computation, Kamel does not disclose how to determine the number of free buffer slots in the write buffer pool.

Finally, Kamel is silent as to *throttling the new write data command if the number of free blocks in the shared read/write buffer pool is insufficient to store the read data for the pending read data requests and the write data for the new write data command*, as recited in limitation (6) of claims 1, 12 and 23. Once again, Kamel does not disclose a shared read/write buffer pool. The Examiner cites col. 7 lines 6-20, col. 6 lines 43-52 and col. 8 lines 40-41 as disclosing this limitation, but the deadline-based queue of read and write requests described in Kamel only prevents insertion of write requests into the queue (and therefore throttles write commands) if the deadlines for the read and write requests already in the queue (based on a prediction of when the write buffer pool will fill up with subsequent *write data*) will not be violated by the insertion of the new write request (col. 8 lines 35-41). In other words, the throttling of write commands in Kamel is based entirely on the expectation of *write data* filling up the write buffer pool. Kamel does not take into account the storing of *read data from pending read data requests*, as in limitation (6).

Because Kamel does not disclose all of the limitations of amended claims 1, 12 and 23, it is respectfully submitted that the rejection of claims 1, 12 and 23 under 35 U.S.C. §102(a) as being anticipated by Kamel has been overcome. In addition, because claims 3-7 depend from claim 1, claims 14-18 depend from claim 12, and claims 25-29 depend from claim 23, the rejection of those claims is traversed for the same reasons provided above with respect to claims 1, 12 and 23.

**Claims 8-11 and 19-22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kamel as applied to claims 1-7, in view of Richter.** This rejection is respectfully traversed.

Claims 8-11 depend from claim 1, and claims 19-22 depend from claim 12. As discussed above, Kamel fails to disclose, teach or suggest limitations (1) through (6), as recited in claims 1 and 12. Furthermore, Richter fails to make up for the deficiencies of Kamel with respect to claims 1 and 12, because like Kamel, Richter also fails to disclose, teach or suggest limitations (1) through (6). In fact, Richter only discloses resource management in very general terms (see FIG. 5 and paragraphs [0211] through [0234].) Richter mentions that when a request for content is received (see paragraph [0214]), the requirements (e.g. resources) for fulfilling that request may be determined (see paragraph [0216]), an indication of whether there are enough available resources may be generated (see paragraph [0219]), and a system monitor may even be told to temporarily slow down on accepting requests for content until the necessary resources become available (see paragraph [0221]). However, there is no detail at all in Richter describing any of the limitations (1) through (6) mentioned above.

Therefore, even if it is assumed that the requisite motivation exists to combine Kamel and Richter, limitations (1) through (6) are completely lacking in both Kamel and Richter.

Because neither Kamel nor Richter, alone or in combination, discloses, teaches or suggests all of the limitations of amended claims 1 and 12, and because claims 8-11 depend from claim 1, and claims 19-22 depend from claim 12, it is respectfully submitted that the rejection of claims 8-11 and 19-22 under 35 U.S.C. §103(a) as being unpatentable over Kamel as applied to claims 1-7, in view of Richter, has been overcome.

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to pass this application to issue.

If, for any reason, the Examiner finds the application other than in condition for allowance, Applicants request that the Examiner contact the undersigned attorney at the Los Angeles telephone number (213) 892-5752 to discuss any steps necessary to place the application in condition for allowance.

In the unlikely event that the transmittal letter is separated from this document and the Patent Office determines that an extension and/or other relief is required, Applicants petition for any required relief including extensions of time and authorizes the Commissioner to charge the cost of such petitions and/or other fees due in connection with the filing of this document to Deposit Account No. 03-1952 referencing Docket No. 491442001600.

Dated: February 24, 2006

Respectfully submitted,

By   
Glenn M. Kubota  
Registration No.: 44,197  
MORRISON & FOERSTER LLP  
555 West Fifth Street, Suite 3500  
Los Angeles, California 90013  
(213) 892-5200